(B) partitioning said determined state at each said program point into components that may each be set separately;

(C) determining the operations required to set each component of the state at each selected program point; and

(D) placing said operations ~~in a way that~~ to eliminates partial redundancies of said operations.

2. (Unamended) The method of claim 1, wherein the data structure stores items on a first-in-last-out basis.

3. (Unamended) The method of claim 2, wherein the states of the data structure are represented as paths on a tree of nodes where:

(A) each path traverses the tree towards the root; and

(B) each node on the path represent a component of the state.

4. (Once Amended) The method of claim 2, wherein the data structure represents actions to be taken by the program if an exception occurs.

5. (Once Amended) The method of claim 4, wherein the selected program points are the points of execution immediately before instructions that might cause an exception.

6. (Unamended) The method of claim 4, further comprising representing the actions to be taken as exception paths in a graph.

7. (New) For a computer-executable program that operates on a data structure, where the data structure must have a required state at selected program points, a method of transforming said program comprising the steps of:

Sub C1 7

(A) analyzing the program to determine the state of an instance of said data structure at said selected program points;

(B) partitioning said instance of said data structure into components;

(C) determining a set of one or more operations for setting the components;

(D) computing placement of the set of operations to eliminate partial redundancies; and

(E) inserting the set of operations at said program points according to the computed placement.

Rule 1.126 11 ~~8~~. (New) The method of claim 7 wherein the data structure is an exception handling stack.

12 ~~9~~. (New) The method of claim 8 wherein the components are a pointer to the exception handling stack and an exception handling data structure.

B2

13 ~~10~~. (New) A machine-readable medium having a set of instructions, which when executed by a set of one or more processors, causes said set of processors to perform operations comprising:

(A) analyzing a program that operates on a data structure, which must have a required state at selected program points in the program, to determine the state of an instance of said data structure at said selected program points;

(B) partitioning said instance of said data structure into components;

(C) determining a set of one or more operations for setting the components;

(D) computing placement of the set of operations to eliminate partial redundancies; and

(E) inserting the set of operations at said program points according to the computed placement.

14 11. (New) The machine-readable medium of claim 10, wherein the data structure stores items on a first-in-last-out basis.

15 12. (New) The machine-readable medium of claim 11, wherein the states of the data structure are represented as paths on a tree of nodes where:

(A) each path traverses the tree towards the root; and

(B) each node on the path represent a component of the state.

16 13. (New) The machine-readable medium of claim 11, wherein the data structure represents actions to be taken by the program if an exception occurs.

17 14. (New) The machine-readable medium of claim 13, wherein the selected program points are the points of execution immediately before instructions that might cause an exception.

18 15. (New) The machine-readable medium of claim 14, wherein the actions to be taken are represented explicitly as exceptional paths in a graph before the transformation, and said exceptional paths are removed.